

MESSAGE-BASED CONVEYANCE OF LOAD CONTROL INFORMATION

The present application claims the benefit of priority of provisional application Serial No. 60/443,573, filed January 30, 2003, the contents of
5 which are incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates to a method and system for controlling processing load in a packet data network, such as an Internet Protocol (IP)
10 Multimedia Subsystem (IMS) provided on top of a packet data network to offer voice and multimedia services e.g. for third generation mobile devices.

BACKGROUND OF THE INVENTION

It is assumed that in the future almost all fixed and mobile
15 communications networks will be based on Internet technology. Especially, services combining several communication types and modes will lead the way in future networks. Voice itself will be just one, although important, piece in the whole communication architecture.

The Session Initiation Protocol (SIP) as defined in the Internet
20 Engineering Task Force (IETF) specification RFC 3261, provides an emerging standard for setting up multimedia sessions on the Internet. Its basic capabilities are setup modification and tear down of any communication session, so it is a signaling protocol. SIP also provides personal mobility, meaning that a consumer is reachable via a single address regardless of its
25 current point of attachment to the network.

In order to support multimedia services, seamless mobility and efficient multiparty conferencing, the IP layer needs to be enhanced. Mobile IP allows terminals to move freely between different mobile networks. SIP is used to establish, modify and terminate sessions. It provides personal mobility
5 by allowing a user to dynamically register to the network with his communication address, i.e. SIP URI (Uniform Resource Indicator). A session is usually a number of Real-time Transport Protocol (RTP) streams to be exchanged. Normally, a session is a combination of speech, audio and video streams, but it may also contain shared applications. A basic SIP network is
10 composed of four types of elements i.e. User Agents (UA), Proxy Servers, Redirect Servers and Registrar Servers. User Agents typically reside in endpoints such as IP phones, personal computers or mobile devices. They initiate requests and provide responses. Usually, UAs also have an interface to media handling and to the actual application software providing the user
15 interface. Proxy servers are intermediaries, which receive and forward requests providing them with, e.g., routing or other services. Redirect servers simply respond to a request by asking its originator to redirect it to a new address. Registrar servers keep track on the actual points of contact of the consumers by accepting registrations from the UAs. Registrar servers and the
20 SIP registration procedure in general provide user mobility as the consumer is able to be reachable from any location via a single address. In this sense, they resemble Home Location Register (HLR) functionality in the Global System for Mobile communications (GSM) networks. Each consumer is part of a domain and each domain runs at least one registrar server, which knows
25 the location of its consumers if they are registered.

SIP uses an address format common to Internet Mail, i.e. "user@domain". The domain part is used to find the correct domain for the consumer and the user part is used to distinguish between individual consumers within a domain. SIP includes request and response messages
5 comprising header fields, e.g. for defining where the request is to be sent next, the recipient address, the sender address etc. Furthermore, a SIP message may contain a payload portion for transmitting subscriber or service specific information.

It is noted that RTP streams do not follow the same path as the SIP
10 message did, but flow directly between the concerned devices. It is thus possible to send the subsequent SIP requests directly between the UAs. In IMS, subsequent SIP messages follow the path recorded into the Record-Route header of the initial request, while interrogating network nodes may drop themselves out and other network elements stay on path. On the other
15 hand, proxy servers in the middle may ask to remain on the signaling path for the duration of the call. This might be useful if the proxy offers some services to the call.

Currently, the Third Generation Partnership Project (3GPP) is specifying IMS e.g. in its specification TS 23.228 as an access independent
20 subsystem which can be used in connection with different networks. IMS uses SIP for session initiation. Basically IMS is just an instance of a SIP network. It has a number of proxies and a registrar. The UA is situated in the terminal device or user equipment (UE). When two devices establish a session they talk to each other via Call State Control Function or Call Session Control
25 Function (CSCF) elements, while RTP media flows do not go via CSCFs but

go directly between the devices. An Application Server (AS) is a SIP element dealing with services, such as advanced call control, presence or instant messaging. The AS may terminate sessions/transactions. The AS may also start sessions/transactions e.g. on behalf of a user or a service.

5 However, there may be situations where the AS does not know whether it should start originating or terminating services when it receives an incoming request message, e.g. an SIP INVITE message, or a serving CSCF (S-CSCF) does not know whether the incoming request message starts an originating or terminating session/transaction. Moreover, other information
10 may be needed for load balancing within the network. Furthermore, for load sharing purposes in a connection processing server (CPS), especially in the S-CSCF and an interrogating CSCF (I-CSCF), it is important to provide a fast and easy algorithm to discover whether a received SIP request is the first in a SIP session or to which SIP session a received request or response belongs
15 to. Currently, SIP does not provide such an efficient means. In order to identify a SIP dialog, i.e. call leg, identified by a combination of call identification, source and destination, a network element or UE has to compare the respective header fields of each SIP message and then to determine whether the call leg already exists. This implies heavy string
20 comparisons and data base queries. A network element which maintains a high number of parallel call legs needs a lot of resources. Additionally, in case of a failure in a network element, information is required about existing sessions.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide an efficient load control scheme for packet data networks.

This object is achieved by a method of controlling processing load in a packet data network, said method comprising the steps of:

- setting a load control information in a predetermined field of a message;
- routing said message through that packet data network;
- checking said load control information on the routing path of said message and/or on the routing path of a response message to said message; and
- 10 - selecting a processing resource of said packet data network in response to the result of said checking step.

Furthermore, the above object is achieved by a network device for controlling processing load in a packet data network, said device comprising: checking means for checking a load control information provided in a predetermined field of a message or response message; and selecting means
15 for selecting a processing resource for said message or response message in response to said checking means.

Additionally, the above object is achieved by a device for transmitting a message to a packet data network, said device being arranged to set into a predetermined field of said message a load control information for selecting
20 processing resources of said packet data network.

Moreover, the above object is achieved by a method of distributing load control information in a packet switched network, comprising the steps of:

- creating a first load control information in a first network element;

- setting said first load control information into a predetermined field of a message;

- routing said message to a second network element;

- storing said first load control information in said second network element;

5 - creating a second load control information in said second network element;

- setting said second load control information into a predetermined field of a second message;

- routing said second message to said first network element; and

- storing said second load control information at said first network element.

10 In addition, the above object is achieved by a system for controlling processing load in a packet data network, said system comprising:

- a first network element for setting a load control information in a predetermined field of a message to be routed in said packet data network; and

15 - a second network element for checking said load control information on the routing path of said message; and for selecting a processing resource of said packet data network in response to the result of said checking step.

Finally, the above object is achieved by a system for distributing load control information in a packet switched network, said system comprising:

20 - a first network element for creating a first load control information and for setting said first load control information into a predetermined field of a message; and

- a second network element for receiving said message, for storing said first load control information, for creating a second load control information, for

25 setting said second load control information into a predetermined field of a

second message, and for routing said second load control information to said first network element;

wherein said first network element is adapted to store said second load control information.

5 Accordingly, load sharing or load balancing information can be routed in messages to the respective network nodes or servers to thereby reduce the amount of resources required for the load control functionality. Moreover, information about existing sessions can be provided in case of failures of network elements.

10 The predetermined field may be a via branch of a SIP message. The load balancing information may be copied from another predetermined field to said predetermined field.

 The predetermined field may be a subfield of a user part of an address header, e.g. a URI of the SIP Route header. Thereby, additional information
15 can be conveyed in the user part. In particular, company confidential information can be carried in one or more subfields e.g., encrypted and/or tokenised and/or encoded information. Furthermore, routing inside a network element, e.g., choosing a correct call state model (for example for originating or terminating session/transaction case) can be conducted using the subfield
20 or subfields of the user part. The load control information and/or other control information may be a subscriber and/or service and/or server specific information carried in one or more subfields e.g., from an S-CSCF to an AS. Alternatively, an IP address may be carried in the subfield, which may be an address of the host in the domain part.

Thus, a plurality of subfields can be provided in the user part for conveying different types of said load control information and/or the other control information. In particular, the user part may be parsed and divided into the subfields. Alternatively, at least one of structure, order and usage of the subfields may be predetermined, e.g., on a standardised basis. The subfields may be separated by a predetermined bit string, character, character string, or other separator.

The selection step may be used for load balancing. A virtual address used for distributing the message to a predetermined processor node. The virtual address may be shared by a plurality of processor nodes. These processor nodes may have a call state control functionality of an IP-based cellular network. Thus, a mechanism to ensure even load balancing is provided, and a subscriber can be tied to as processor node for the whole registration period. By using a virtual address, a cluster itself can do load balancing for cluster nodes.

Additionally, the load control information may comprise a first port number indicating a first port for receiving a request message. Furthermore, the load control information may comprise a second port number indicating a second port for receiving a response message. Thereby, requests and responses can be received at the indicated port where a load balancing information is provided.

In another aspect, the load control information may comprise a first information and an optional second information. The first and optional second information may be set in a header field selected from a Route header field, Via header field, or Contact header field of a SIP message. The first

information may indicate whether a session of the message already exists. The optional second information may then indicate an identification of the existing session. The first and second information may be a hidden information not meaningful to other networks.

5 Accordingly two alternatives can be provided. In the first alternative, it is only detected whether the message is the first one in a call leg. Thereby, easy and fast recognition of the first message in a session can be provided. No change to other network elements or terminals is required. The scheme may even be provided on a non-standardised basis. In the second alternative,
10 an additional session identification is detected based on the second information. Thereby, in addition to the above advantages of the first alternative, easy and fast recognition of the session in question can be provided.

 In particular, the first and/or second information may be set as a part of
15 user part in an address (e.g. SIP URI) of a header field, as a part of host name of a header field, as a part of domain name of a header field, as a parameter of the header field, as a port number of the header field, wherein the port number may be used for differentiating a first message from subsequent messages, or as an extension header field to the message.
20 Alternatively, the first and/or second information may be set in a payload portion of the message.

 The second information may then be extracted in response to a detection of the first information, and may be used in the selection step.

 The network device for controlling the processing load may comprise a call
25 state or call session control functionality of an IP-based cellular network. The

selecting means may be arranged to select a predetermined processor node to which said message is distributed. Thus, in addition to the virtual address, the load control information specifies the processor node.

5 The selecting means may be arranged to initiate creation of a new session. The device for transmitting the message may also comprise call state control functionality or call session control functionality of the IP-based cellular network. This session control functionality may be a serving call session control functionality, an interrogating call session control functionality or a proxy call session control functionality. The device may be arranged to set the
10 load control information in the user part of the header address of the message, or, alternatively, in a host name, domain name, header parameter, port number, or extension header field of a header portion or in a payload portion of the message.

Other advantageous developments of the present invention are defined
15 in the dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will now be described on a basis of preferred embodiments with reference to the accompanying drawings, in which:

20 Fig.1 shows an IMS network architecture in which the present invention can be implemented;

Fig. 2 shows a structure of an SIP URI according to the first preferred embodiment;

Fig. 3 shows a signalling and processing diagram of a first signalling
25 example according to the first preferred embodiment;

Fig 4 shows a processing and signalling diagram indicating a second signalling example according to the first preferred embodiment;

Fig. 5 shows a signalling and processing diagram according to a second preferred embodiment;

5 Fig. 6 shows a flow diagram of a first example of a load sharing mechanism according to the second preferred embodiment; and

Fig. 7 shows a flow diagram of a second example of a load sharing mechanism according to the second preferred embodiment.

10 DESCRIPTION OF THE PREFERRED EMBODIMENT

The preferred embodiments will now be described on the basis of an IMS network architecture as shown in Fig. 1.

It is noted that Fig. 1 only shows those IMS components required for a description of the present invention. As an example, the radio access network and the core network are not shown in Fig. 1. According to Fig. 1, a terminal
15 device or UE 10, which may be a mobile or cellular terminal device, is connected to a P-CSCF 20 arranged in a visited domain 70 of the UE 10 and providing basic IP connectivity and mobile management below it. The UE 10 uses SIP to communicate with the P-CSCF 20 which is similar to a SIP proxy
20 server. In the present case, the consumer or subscriber of the UE 10 is roaming in the visited domain 70 where the P-CSCF 20 is located. The role of the P-CSCF 20 is to provide emergency call and other such services requiring specific knowledge of the visited domain 70. Instead of radio access network also alternative access networks may be used. Instead of mobile or cellular

terminal device also other kind of terminals may be used, especially in alternative access networks.

Furthermore, an S-CSCF 40 is always located in the subscriber's or consumer's home domain 80 and takes the role of the SIP registrar and proxy servers, so that the UE 10 can be registered at the S-CSCF 40 using SIP via the P-CSCF 20. Furthermore, an I-CSCF 30 is provided as another SIP proxy server responsible mainly for finding the correct S-CSCF for a given subscriber or consumer. The S-CSCFs can be dynamically allocated per registration in order to achieve efficient load balancing and error residency. An Application Server (AS) 60 is provided as a SIP element dealing with the services provided to the UE 10. Separate ASs can be provided for different purposes. Finally, a Home Subscriber Server (HSS) 50 is arranged for profile management and authentication.

In the following, a first preferred embodiment of the present invention will be described on the basis of Figs. 2 to 6.

In the first preferred embodiment, a content of a user part of SIP URI is used for load control, e.g., session control and load balancing. In particular, the user part of the SIP URI (Uniform Resource Identifier) can be divided into subfields which may be utilized e.g. for control and/or routing purposes. In SIP, the Route Header normally contains SIP URIs which have only a domain part, such that the user part is free to be used for other purposes.

Fig. 2 shows a schematic diagram indicating a structure of the FQDN or SIP URI 100 according to the first preferred embodiment. The SIP URI 100 comprises a user part 120 and a domain part 140 separated by an "@" sign, similar to an e-mail address. The objects addressed by SIP are users at

hosts, identified by the SIP URI 100. The user part 120 is used for carrying a user name or a telephone number, while the host or domain part 140 carries either a domain name or a numeric network address.

Due to the fact that the user part is not used in the Route header
5 neither in the Via header, it can be divided into subfields 121, 122, ...12n, which may be separated by a suitable separator 130, e.g. a bit string, character or character string, wherein the characters can be printable and/or non-printable characters or bit strings. The order and usage of the subfields 121 to 12n can be predetermined or standardized if not considered as an
10 implementation specific information.

Regarding the arrangement and structure of the subfields 121 to 12n in the user part 120, three options may be used. According to the first option, the user part 120 may be arranged as a single field, which contains the subfields 121 to 12n. This single field is then passed and divided into the subfields 121
15 to 12n, when needed. This provides the advantage that no standardization is needed if the field is created and utilized inside the same network.

According to the second option, the user part 120 may structurally consist of the subfields 121 to 12n, while the syntax and possibly also the semantics of the subfields 121 to 12n are predefined or standardized. In this
20 case, the subfields 121 to 12n can be created and utilized even in different networks.

According to the third option, a combination of the first and second option may be used.

The following is an example of the SIP URI 100 where a second
25 subfield is used for signaling the session/transaction case and a first subfield

is used for signaling an implementation specific information. The separator 130 is formed by the corrector "_":

57BC442C_terminating@s-cscf2.ims.sonera.fi

Accordingly, "terminating" is signaled as the session/transaction case and

5 "57BC442C" is signaled as the implementation-specific information.

In the following, first and second examples of a load control mechanism according to the first preferred embodiment are described with reference to Figs. 3 and 4.

The load control mechanism is provided to ensure even load balancing
10 if a network element or part is implemented by a number of processor nodes. In the IMS network architecture according to Fig.1, the UE 10 has the P-CSCF 20 as a contact point to the network. Between the UE 10 and the P-CSCF 20, an IP security function IPSec is used for integrity and confidentiality protection. Furthermore, a compression function may be used to compress
15 the signaling information in the prefix or user part 120 of the FQDN or SIP URI 100 of the UE 10. In order to achieve high capacity and fast response times, the IPSec data and compression data of the particular subscriber of the UE 10 must be stored in a memory e.g. volatile memory or random access memory (RAM). As a result a subscriber should be tied to the processor node
20 to which it is registered. By using the virtual address, only the node or server cluster itself can do load balancing for cluster nodes.

Fig. 3 shows a signaling and processing diagram of a load control mechanism for distributing a load balancing information (LBI) upon registration of a user. Here, reasoning for load balancing is that e.g.
25 compression is done at distributed nodes. Therefore, in practice at the

interface between the UE 10 and the P- CSCF 20 load balancing cannot be done based on SIP level information since it is compressed. At this interface reasonable load balancing key is the IP-address of the UE 10. When a terminating attempt or request is received, which is targeted to the UE 10 it is essential that the terminating attempt is received at the same processing node to which messages from the UE 10 are distributed. Thereby, unnecessary hops can be avoided in the IP-based network.

When the UE 10 transmits in step 1 a SIP Register message, the P-CSCF 20 creates and inserts in step 2 a first load balancing information LBI(P-CSCF-1) to SIP-URL(P-CSCF) of the Path field of the header of that Register message. The first load balancing information LBI(P-CSCF-1) in the Path field will be received in future when an initial request is received from the S-CSCF 40. The P-CSCF 20 also creates and inserts a second load balancing information LBI(P-CSCF-2) to the Via branch at that Register message. The second load balancing information LBI(P-CSCF-2) at the Via branch will be received along with a response to that Register message. The first and second load balancing information LBI(P-CSCF-1) and LBI(P-CSCF-2) may be identical or different. The register message is routed in step 3 and 4 via the I-CSCF 30 to the S-CSCF 40. When the S-CSCF 40 receives the Register message from the P-CSCF 20, it does load balancing in step 5 based on e.g. the Call ID. In step 6, the S-CSCF 40 then stores to a subscriber database the SIP-URL(P-CSCF) from the Path field, which contains the first load balancing information LBI(P-CSCF-1) of the P-CSCF 20. In step 7, the S-CSCF 40 creates an own load balancing information LBI(S-CSCF-1) and inserts it to the SIP-URL(S-CSCF) of the Service-Route

field of the SIP 200OK response message sent in steps 8 and 9 via the I-CSCF 30 to the P-CSCF 20. This load balancing information LBI(S-CSCF-1) will be received in future when an initial request is received from the P-CSCF 20. In addition, the Via branch contains the SIP-URL(P-CSCF) of the P-CSCF 20 which has been copied from the Register message received after step 4. When the P-CSCF 20 receives this 200OK response message, load balancing can be done in step 10 based on the second load balancing information LBI(P-CSCF-2) obtained from the Via branch. In step 11, the P-CSCF 20 stores in a database the SIP-URL(S-CSCF) containing the load balancing information LBI(S-CSCF-1) of the S-CSCF 40 obtained from the Service Route field of the 200OK response message. Finally, in step 12, the 200OK response message indicating successful registration is forwarded to the UE 10.

Accordingly, after the above procedure, the P-CSCF 20 has the SIP-URL(S-CSCF) at its subscriber data which points to the S-CSCF 40 and which contains the corresponding load balancing information LBI(S-CSCF-1). Similarly, the S-CSCF 40 has the SIP-URL(P-CSCF) at its subscriber data which points to the P-CSCF 20 and which contains the corresponding load balancing information LBI(P-CSCF-1). The load balancing information can thus be used later by the respective load balancers of the P-CSCF 20 and the S-CSCF 40. For example, when a terminating attempt happens, the S-CSCF 40 then fetches this load balancing information from the subscriber database and inserts it to the corresponding request.

Fig. 4 shows a processing and signaling diagram of a load control mechanism for using the load balancing information (LBI) when an initiation

request is sent to the network. When the UE 10 sends in step 1 a SIP Invite message to the P-CSCF 20, load balancing is done based on the IP address of the UE 10. In step 2, the P-CSCF 20 reads from the subscriber database the previously stored SIP-URL(S-CSCF) to be used for routing the Invite message to the S-CSCF40 in step 3. Additionally, the SIP-URL(S-CSCF) is inserted to the topmost Route field, and the SIP-URL(P-CSCF) is inserted to the Record-Route field. Moreover, the first load balancing information LBI(P-CSCF-1) is inserted to the Via branch. When the Invite message is received at the S-CSCF 40, it obtains from the topmost Route field the SIP-URL(S-CSCF) which contains its previously set load balancing information LBI(S-CSCF-1). Based on this load balancing information LBI(S-CSCF-1) load balancing is done in step 4 to find a correct computer. When the S-CSCF 40 sends the Invite message in step 5, it inserts the SIP-URL(S-CSCF) to the Record-Route field and inserts its load balancing information LBI(S-CSCF-1) to the Via branch. In step 6, the application server 60 responds with a 200OK response message comprising the load balancing information LBI(P-CSCF-1) and LBI(S-CSCF-1) of the P-CSCF 20 and the S-CSCF 40 in the Via branch. When the S-CSCF 40 receives the 200OK response message, it obtains its load balancing information LBI(S-CSCF-1) from the Via branch and uses it for load balancing in step 7. Similarly, when the P-CSCF 20 subsequently receives the 200OK response message forwarded by the S-CSCF 40 in step 8, it obtains its first load balancing information LBI(P-CSCF-1) from the Via branch and uses it for load balancing in step 9. In step 10, the 200OK message is forwarded the UE 10 to acknowledge the previous Invite message.

When the application server 60 sends in step 11 a subsequent SIP request, e.g., an Invite message, within a dialog, it uses a Route list which it has previously created based on the Record-Route entries of the initial request, i.e. Invite message received after step 5. The topmost Route entry is the SIP-URL(S-CSCF) including the corresponding load balance information LBI(S-CSCF-1) therein. The second Route entry is the SIP-URL(P-CSCF) including the corresponding load balance information LBI(P-CSCF-1) therein. When the S-CSCF 40 receives the subsequent Invite message, it obtains from the topmost Route entry its load balancing information LBI(S-CSCF-1) which is inside the SIP-URL(S-CSCF). In step 12, the S-CSCF 40 does load balancing based on this load balancing information LBI(S-CSCF-1) and removes the Route entry pointing to itself. Now, the topmost Route entry points to the P-CSCF 20. In step 13, the subsequent Invite message is forwarded to the P-CSCF 20. When the P-CSCF 20 receives the subsequent Invite message, it obtains from the topmost Route entry its load balancing information LBI(P-CSCF-1) which is provided inside the SIP-URL(P-CSCF). Then, it does load balancing based on this load balancing information LBI(P-CSCF-1) in step 14, and removes the Route entry pointing to itself. Finally, in step 15, the Invite message is forwarded to the UE 10.

In general, at the registration phase the path and load balancing information is recorded and stored to be used later for requests. The later request should contain this load balancing information and load balancing is then done based on that load balancing information. Accordingly, any load balancing information inserted during the registration phase is for future requests.

In all cases, the via-branch parameter of the Via header field may be used to carry similar information used by the load balancing function to distribute responses to the correct processor node.

Furthermore, different port numbers may be used to identify where the
5 load balancing information can be found. In particular, during “path recording”, the request port is set to the domain part 140 of the SIP URI 120. Thus, requests are then received at that request port and it is known where to read the load balancing information. Similar, “port” can be set for out going requests, such that responses are received at that port.

10 In case of a load balancing function for an ingress or incoming SIP traffic which is destined to the concerned network element, the destination IP address is checked whether it is a virtual IP address or not. If not, load balancing is not needed, e.g. normal L3 routing can then be applied for packet. If the destination IP address is a virtual IP address, then load
15 balancing is needed. There are two choices, i.e., the virtual IP address can be either a P-CSCF address or it can be an S-CSCF or I-CSCF address. A corresponding destination port information is used to detect the kind and location of the load balancing information. Then, load balancing is done based on the load balancing information and the resulting output corresponds to the
20 correct processing node to where packets should be forwarded. The load balancing information may be a call id, UE-IP-address, or a previously generated load balancing information.

In case of an egress or outgoing traffic which is originated from the concerned network element, the source IP-address is checked whether it is
25 one of the virtual IP-addresses (P-CSCF, S-CSCF or I-CSCF) of the CPS. If

not, then normal routing happens. If yes, it is checked if it is a Loop address. If so, the transport protocol is determined and subsequently the destination port is checked to determine a request port or dedicated port. Based on the checking result a procedure for obtaining the load balancing information and forwarding the IP packet is selected. In case of a Not-Loop Address, the source IP-address is again checked to determine whether it is an S-CSCF/I-CSCF address or a P-CSCF address. If it is an S-CSCF/I-CSCF address, the transport protocol, i.e. Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), is determined. If TCP is indicated, the SIP message is reassembled after buffering and then forwarded. If UDP is indicated, the IP packet can be directly forwarded. If it is P-CSCF address, the source port is determined. If a Client port or Request port is indicated, the transport protocol is determined and the above processing is again initiated. In case a UE unsecure/secure client/server port is indicated, the IP packet is directly forwarded by an L3 (Protocol Layer 3) procedure.

In the following, the second preferred embodiment of the present invention is described with reference to Figs. 5 to 7. The second preferred embodiment is directed to a load sharing mechanism for discovering in an efficient way which SIP traffic belongs to which session, and whether a request, e.g. a SIP INVITE request, is an initial request or a re-request.

Fig. 5 shows a processing and signaling diagram indicating a first example of the load sharing mechanism according to the second preferred embodiment. The mechanism of the first example is adapted to detect, whether a SIP request, e.g. SIP INVITE is the first one in a call leg. This is

achieved by providing a hidden information in the Record-Route header field of the SIP request.

Whenever a session-stateful CSCF, e.g. the S-CSCF 40 in Fig. 1, receives a SIP request and inserts a Record-Route header field (step 1), it
5 inserts a hidden indication into the Record-Route header field (step 2), and forwards the SIP request with the hidden indication towards the destination address. In the present case, "hidden" means that the indication is not meaningful to other networks, e.g. networks outside the network where the indication is set. However, if required, the added indication may as well be a
10 standardized information readable by other networks.

Then, whenever an INVITE arrives, a session-stateful CSCF checks, whether the topmost Route header field or request URI (if there is no Route header) contains such a hidden information. The Route header field is constructed from the Record-Route header field. If the hidden information or
15 indication is present, the session is already existing. If not, a new session has to be created internally in the concerned session-stateful CSCF.

As SIP responses (e.g. 200 OK) in normal case belong to an existing session, they don't need to be distinguished.

The indication can be part of the hostname. As an example, it is
20 assumed the default routing address to this element would be <scscf17.operator.net> such as <B@provider.net; maddr=scscf17.operator.net>, the Record-Route could then look like:
RecordRoute: <B@provider.net; maddr=exist.scscf17.operator.net>
or
25 *Record-Route: <B@exist.provider.net>*

or

Record-Route: <B@exist.scscf17.operator.net>.

Here the hidden indication would be "exist." as part of the hostname. The user part 120 may be empty in these examples. E.g., instead of

5 *<B@exist.scscf17.operator.net>* the following may be used: *<ex-
ist.scscf17.operator.net>.* The Route header field is constructed from the Record-Route header field. For example:

Record-Route: <B@exist.scscf17.operator.net>

gives

10 *Route: <B@exist.scscf17.operator.net>.*

As an alternative, the S-CSCF 40 may add an "rr-param" to the Record-Route header field as defined in RFC3261:

*Record-Route = "Record-Route" HCOLON rec-route *(COMMA rec-route)*

*rec-route = name-addr *(SEMI rr-param)*

15 *rr-param = generic-param*

generic-param = token [EQUAL gen-value]

gen-value = token / host / quoted-string

token = 1(alphanum / "-" / "." / "!" / "%" / "*" /
/ "_ " / "+" / "\" / "" / "~")*

20 *Route = "Route" HCOLON route-param *(COMMA route-param)*

*route-param = name-addr *(SEMI rr-param)*

A corresponding example may look as follows:

RecordRoute: <B@provider.net; maddr=scscf17.operator.net; existing>

or

25 *Record-Route: <B@provider.net; existing>*

or

Record-Route: <B@scscf17.operator.net; existing>.

Again, the user part 120 may be empty. The Route header field is constructed from the Record-Route header field.

5 If the rr-param "existing" is received, it can easily be detected that the request belongs to an existing session.

According to a further alternative, different port numbers may be used for the first and for the subsequent requests. As an example, it is assumed that the default routing to the concerned element or node would be

10 <B@provider.net; maddr=scscf17.operator.net>. Then, the RecordRoute entry could look like :

RecordRoute: <B@provider.net; maddr= scscf17.operator.net:19373>

or

Record-Route: <B@provider.net:19373>

15 or

Record-Route: <B@scscf17.operator.net:19373>.

And here again, the user part 120 may be empty. The Route header field is constructed from the Record-Route header field. For example

Record-Route: <B@provider.net; maddr= scscf17.operator.net:19373>

20 gives

Route: <B@provider.net; maddr= scscf17.operator.net:19373>.

All the requests, which arrive at port 19373, would be recognized as belonging to an existing session.

According to another alternative, a new proprietary or non-standardized extension header field may be used in SIP to convey the information. An example of the new header entry may look as follows:

CSCF-session: existing in scscf17.operator.net

- 5 However, in this case, the UA would have to support this feature, i.e. copying this new header field from the SIP requests to the SIP responses and subsequent SIP requests.

According to a still other alternative, the payload portion of the SIP request could be used for conveying the hidden indication.

- 10 Fig. 6 shows a schematic flow diagram indicating the load sharing mechanism according to the first example. When an initial request is handled by the element, Record-Route header can be inserted to the request before sent out. To this Record-Route header a hidden information can be added. Later these Record-Route headers are passed back within response to the
- 15 originator of the request. When the originator gets this response, it gets those Record-Route headers and copies them to the Route list. When the originator sends subsequent request, it takes this Route list and inserts all entries to the subsequent request as Route headers. Now subsequent request can be sent out. When the next server receives this request, it can find from the Route
- 20 header same SIP URI it inserted earlier to initial request.

A Via header is also inserted to the request. Same Via header is received in the response to the request, and hidden information in the Via header can be used to find the instance or element to which the response must be passed.

Thus, according to Fig. 6, if a SIP request is received, the Route header field, or new header field or payload portion, is checked for the hidden indication (step S201). If a hidden indication is detected in step S202, a session is already existing and no creation is required and the request can be allocated to the call leg of the existing session (step S204). On the other hand, if no hidden indication is detected, a new session is created in step S203.

According to the second example of the load sharing mechanism, an internal session identifier (ISId) is detected based on the hidden indication. This alternative includes the above mechanism of the first example, i.e. if the ISId cannot be discovered, we can assume, that the request belongs to a new call leg.

In the second example, a hidden indication is provided e.g. in the Record-Route header field and in the Via header field of the SIP request. Thus, referring to Fig. 5, whenever a session-stateful CSCF, e.g. the S-CSCF 40, inserts a Record-Route or Via header field, it will add a hidden indication, containing information about the internal session identifier (ISId).

Fig. 7 shows a schematic flow diagram indicating the enhanced load sharing mechanism according to the second example. Whenever a message arrives, it is checked whether the message corresponds to a SIP request (step S301). If so, the session-stateful CSCF checks in step S303 whether the topmost Route header field contains a hidden indication. If it is determined in step S305 that the hidden indication is present, the session is existing and the ISId can be extracted to provide a fast session recognition and allocation function (step S307). If not, a new session has to be created in step S306.

If no SIP request is detected in step S301, it is checked in step S302 whether the message corresponds to a SIP response. If so, the session-stateful CSCF checks in step S304 whether the topmost Via header field contains a hidden indication. If it is determined in step S305 that the hidden
5 indication is present, the session-stateful CSCF extracts the ISId from the topmost Via header field of the SIP response (step S307).

Hence, even a quick identification of an existing session can be provided at the CSCFs.

As in the first example, the hidden indication can be part of the
10 hostname. As an example, it is assumed that the default routing to this element would be <scscf17.operator.net> such as <B@provider.net; maddr=scscf17.operator.net>. Then, the Route header field could look like the following. The Route header field is constructed from the Record-Route header field.

15 *Route: <B@provider.net; maddr=ISId224497.scscf17.operator.net>*

or

Route: <B@ISId224497.provider.net>

or

Route: <B@ISId224497.scscf17.operator.net>.

20 Here, the ISId is "224497" as part of the hostname. The user part 120 may be empty.

Similarly, the Via header field could be used, which could then look like:

Via: SIP/2.0/UDP ISId224497.scscf17.operator.net:5060

All the SIP responses which contain "ISId224497" as part of the hostname,
25 belong the same existing session.

As an option, the ISId as part of the hostname could also be encrypted/tokenized for hiding or redundancy purposes.

As an alternative, the S-CSCF 40 may add a "rr-param" to the Record-Route header field as defined in RFC3261.

- 5 Similarly, this works for SIP responses, where the Via header field would be extended by a "via-extension" as defined in RFC3261:

Via = ("Via" / "v") HCOLON *via-parm* *(COMMA *via-parm*)

via-parm = *sent-protocol* LWS *sent-by* *(SEMI *via-params*)

via-params = *via-ttl* / *via-maddr*

10 / *via-received* / *via-branch*
/ *via-extension*

via-ttl = "ttl" EQUAL *ttl*

via-maddr = "maddr" EQUAL *host*

via-received = "received" EQUAL (*IPv4address* / *IPv6address*)

15 *via-branch* = "branch" EQUAL *token*

via-extension = *generic-param*

sent-protocol = *protocol-name* SLASH *protocol-version*
SLASH *transport*

protocol-name = "SIP" / *token*

20 *protocol-version* = *token*

transport = "UDP" / "TCP" / "TLS" / "SCTP"
/ *other-transport*

sent-by = *host* [COLON *port*]

ttl = 1*3DIGIT ; 0 to 255

25 *generic-param* = *token* [EQUAL *gen-value*]

gen-value = *token* / *host* / *quoted- string*
token = 1*(*alphanum* / "-" / "." / "!" / "%" / "*" /
 / " _ " / "+" / "`" / "'" / "~")

5 As an example, the Route header field could then look like the following. The Route header field is constructed from the Record-Route header field.

Route: <B@provider.net; maddr=scscf17.operator.net; ISId=224497>

or

10 *Route: <B@provider.net; ISId=224497>.*

or

Route: <B@scscf17.operator.net; ISId=224497>.

Here, the ISId is "224497" as parameter of the Route header field.

The corresponding Via header field could then look like:

15 *Via: SIP/2.0/UDP scscf17.operator.net:5060; ISId=224497.*

In this case, all the SIP responses which contain the parameter "ISId=224497" belong to the same existing session.

Optionally, the ISId as part of the hostname could also be encrypted/tokenized for hiding or redundancy purposes.

20 According to a further alternative, different port numbers may be used for all existing sessions:

As an example, it is assumed that the default routing to the concerned element would be <B@provider.net; maddr=scscf17.operator.net>, the Route header field could then look like the following. The Route header field is constructed from the Record-Route header field.

25

Route: <B@provider.net; maddr= scscf17.operator.net: 224497>

or

Route: <B@provider.net: 224497>

or

5 *Route: <B@scscf17.operator.net: 224497>.*

User part may be empty. All the SIP requests, which arrive e.g. at port 224497, then belong the same existing session.

Similarly, this works for the Via header field, which could look like:

Via: SIP/2.0/UDP scscf17.operator.net: 224497

10 Here, all the SIP responses, which arrive at port 224497, belong the same existing session.

However, listening to many ports in parallel might cause performance or scalability problems.

As another alternative, a new proprietary extension header field may be
15 used in SIP to convey the hidden information. As an example, the new extension header field may look like:

CSCF-session: "ISId=224497 in scscf17.operator.net"

However, as already mentioned in connection with the first example, the UA
would have to support this feature, i.e. copying this new header field from the
20 SIP requests to the SIP responses and subsequent SIP requests.

As a still further alternative, the payload portion of the SIP request or response could be used for this too.

Mainly, the above load sharing mechanisms are provided in CSCFs or other network nodes with corresponding functionality. However, in general,
25 they may also be implemented in terminal devices, such as the UE 10.

It is noted that the present invention is not restricted to the preferred embodiments described above. The present invention may be implemented in any network node having a load control functionality in any network. In particular, any header or payload fields of any packet data
5 message or datagram may be used. Furthermore, any information usable for load control can be conveyed. The embodiments may thus vary within the scope of the attached claims.